

Directions Complete the exercises. Your solutions to the exercises should be submitted to Gradescope before the indicated due date above. Please follow rules regarding Gradescope submission as described in the syllabus.

References Except for the help of the instructor or TAs and the class textbooks and notes, if you use any resources, for example, a book, a website, or you discussed with your friends, please acknowledge them in this References section.

- I discussed Problem ?? with STUDENT A, STUDENT B, ...
- I used BOOK/WEBSITE to help me do Problem ??.

Exercises

1. Look at the mathematical model for the growth of the red wolf population, when wild born and introduced wolves were considered in the last homework.

- (a) In addition to the wild red wolf population there is a captive breeding population. One strategy is to introduce captive bred wolves into the wild. Suppose that 10 captive bred wolves are added to the wild population each year. Unfortunately, captive bred wolves do not readily integrate into existing packs, and are less effective hunters than wild-born wolves. As a result 43% of introduced wolves die each year. To model the effect of introducing wolves, we must keep track of both the number of wild-born wolves, N_t and the number of wolves that are introduced I_t .

- (i) Explain why:

$$N_{t+1} = 1.06N_t + 0.28I_t$$

$$I_{t+1} = 10 + 0.57I_t$$

(explain what each term in the equations represents). Also explain why the total number of wild red wolves is given by $N_t + I_t$

Given the number of wild wolves at time t is N_t and the number of captive bred wolves at time t is I_t , the total number of wolves would be the sum of the two populations i.e $N_t + I_t$. Let $b_c = 0.28$ be the birth rate of the captive bred wolf population. Since the $0.28I_t$ wolves born at time t are born in the wild, they will be a part of the wild wolf population at time $t + 1$. Let $R_0 = b_w - m_w = 0.06$ be the net reproduction rate of the wild wolf population. Thus, the number of wild wolves at time $t + 1$ (N_{t+1}) is the sum of the number of wolves at time t (N_t), the number of wild wolves born from wild parents minus the number of wild wolves that died at time t ($R_0N_t = (b_w - m_w)N_t$), and the number of wild wolves born from captive bred parents (b_cN_t). Let $m_c = 0.43$ be the mortality rate for captive bred wolves. Let $r = 10$ be the number of captive bred wolves introduced each year. Thus, the number of captive bred wolves at time $t + 1$ (I_{t+1}) is the sum of the number of captive bred wolves at time t minus the number of captive bred wolves that died at time t ($(1 - m_c)I_t$) and the number of captive bred wolves introduced at time t (r).

- (ii) Calculate N_t and I_t for $t = 0, 1, 2, 3, 4, 5$, assuming that $N_0 = 130$ and $I_0 = 0$. Then, using Matlab, or by some other means, calculate when the total number of wild red wolves will

reach the target population size of 220.

The population will first the 220 wild wolf threshold at $t = 7$.

Out[35]:

	Time (Years)	N_t	I_t
0	0.0	130.000000	0.000000
1	1.0	137.800000	10.000000
2	2.0	148.868000	15.700000
3	3.0	162.196080	18.949000
4	4.0	177.233565	20.800930
5	5.0	193.691839	21.856530

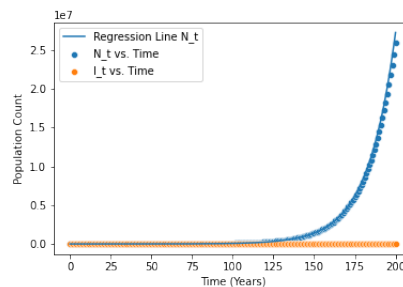
```
In [36]: def red_wolf_population_goal(N_0, I_0, goal_pop):
t=0
while N_0 < goal_pop:
    N_0 = 1.06 * N_0 + 0.28 * I_0
    I_0 = 10 + 0.57 * I_0
    t += 1
return N_0, t
```

```
In [37]: red_wolf_population_goal(N_t, I_t, 220)
```

Out[37]: (230.40747073691688, 7)

- (iii) Make a plot of how the population size grows as a function of t . Show that the growth is eventually exponential.

As t grows large, the plot of N_t vs. time appears to closely follow the exponential regression



graph.

- (b) Explain why the recurrence equations that you derived for I_t and N_t can not be written as an iterated linear map (i.e. using a Leslie matrix) - identify the specific term(s) that can not be represented using the Leslie matrix.

Every generation, I_{t+1} depends a constant number of captive bred wolves added to the population. This constant quantity is independent of I_t and N_t and therefore can't be expressed in Leslie matrix form.

- (c) For the Leslie matrix:

$$L = \begin{pmatrix} 1 & 3/2 \\ 2 & 1/2 \end{pmatrix}$$

- (i) Assume that you have initial data: $\mathbf{N}^{(0)} = \begin{pmatrix} 100 \\ 200 \end{pmatrix}$. Calculate by hand, or using a computer, $\mathbf{N}^{(k)}$ for $k = 1, 2, 3$.

	time	N_0	N_1
0	0.0	100.0	200.0
1	1.0	400.0	300.0
2	2.0	850.0	950.0
3	3.0	2275.0	2175.0

(ii) Verify that $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is an eigenvector of the Leslie matrix, and find the corresponding eigenvalue.

Then derive a formula for the subpopulation sizes at arbitrary k if $\mathbf{N}^{(0)} = \begin{pmatrix} 100 \\ 100 \end{pmatrix}$.

$\begin{pmatrix} 1 & 3/2 \\ 2 & 1/2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix}$ Thus, 2.5 is an eigenvalue, and $\mathbf{N}^{(k)} = (2.5)^k \mathbf{N}^{(0)}$

(d) For the Leslie matrix:

$$L = \begin{pmatrix} 11/2 & 3/2 \\ 3/2 & 3/2 \end{pmatrix}$$

find both of the eigenvalues and the corresponding eigenvectors. Explain why there is at least one initial condition for which a model using this matrix would predict unbounded exponential growth.

$$\det(L - x \cdot I) = (6 - x)(1 - x) = 0 \text{ for } x = 6, 1$$

$$L \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 18 \\ 6 \end{pmatrix}$$

$$L \begin{pmatrix} -1 \\ 3 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \end{pmatrix}$$

There exists a dominant eigenvalue $\lambda = 6$, so the sub-populations will grow exponentially as t becomes large by a factor of 6 asymptotically approaching the line in the direction of the $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ vector.

(e) For the Leslie matrix:

$$L = \begin{pmatrix} 5 & 1 & 1 & 0 \\ 3/2 & 1/2 & 0 & 1 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \end{pmatrix}$$

show, without directly computing the eigenvalues of L , that there is an eigenvalue larger than 1, so the population will grow exponentially.

$$\det(L - 1 \cdot I) = \frac{-15}{4} < 0$$

Because L 's characteristic polynomial is 4th degree with a leading coefficient of 1. $(5 - x)(1 - x)(-x)(-x) = x^4 + \dots$ is the only diagonal contributing to the fourth degree term. Since $\det(L - 1 \cdot I) < 0$ and $\det(L - xI)$ is increasing for all $x > 1$, $\det(L - xI)$ must have a root for some $x > 1$ by the IVT.

```

In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: M_t=130
I_t=0

In [3]: def red_wolf_population_df(M_0,I_0,time):
df=pd.DataFrame({
    "Time (Years)":i[0],
    "M_t":(M_0),
    "I_t":(I_0)}
    for i in np.arange(1,time+1):
        M_0+=M_0*0.22*I_0
        I_0+=I_0*0.57*I_0
        x=("Time (Years)"+"M_t"+M_0,"I_t"+I_0)
        df=df.append(x,ignore_index=True)
    return df

In [4]: df=red_wolf_population_df(130,0,5)

In [5]: df

Out[5]:
```

	Time (Years)	M_t	I_t
0	0.0	130.000000	0.000000
1	1.0	137.800000	10.000000
2	2.0	148.868000	15.700000
3	3.0	162.196080	18.949000
4	4.0	177.233565	20.80093
5	5.0	193.691839	21.85653

```

In [6]: def red_wolf_population_goal(M_0,I_0,goal_pop):
t=0
while M_0<goal_pop:
    M_0+=M_0*0.22*I_0
    I_0+=I_0*0.57*I_0
    t+=1
return M_0,t

In [7]: red_wolf_population_goal(M_t,I_t,220)

Out[7]: (230.40747073691688, 7)

In [8]: df_graph=red_wolf_population_df(130,0,200)

In [9]: M_t_exp=np.exp(np.polyfit(x=df_graph["Time (Years)"][df_graph["I_t"]>22.255],y=np.log(df_graph["M_t"][df_graph["I_t"]>22.255]),deg=1))

In [10]: sns.lineplot(data=df_graph,x="Time (Years)",y=M_t_exp[1]*np.power(M_t_exp[0],df_graph["Time (Years)"]),label="Regression Line M_t")
sns.scatterplot(data=df_graph,x="Time (Years)",y="M_t",label="M_t vs. Time")
sns.scatterplot(data=df_graph,x="Time (Years)",y="I_t",label="I_t vs. Time")
plt.ylabel("Population Count")
plt.savefig("Math_142_Homework_2_Q_1_a")

In [11]: np.corrcoef(x=df_graph["Time (Years)"],y=np.log(df_graph["M_t"]))

Out[11]: array([[1., 0.9996912],
[0.9996912, 1.]])

In [12]: L_c=np.array([[1,3/2],[2,1/2]])

In [13]: M_c=np.array([[100],[200]])

In [14]: np.matmul(L_c,M_c)

Out[14]: array([[400.],
[300.]])

In [15]: def leslie_calc(L,M_0,time):
df_L=pd.DataFrame({
    "time":i[0],
    "M_0":(M_0[0][0]),
    "M_1":(M_0[1][0])})
for i in np.arange(1,time+1):
    M_0=np.matmul(L,M_0)
    x=("time"+i,"M_0"+M_0[0][0],"M_1"+M_0[1][0])
    df_L=df_L.append(x,ignore_index=True)
return df_L

In [16]: df_L_c=leslie_calc(L_c,M_c,3)
df_L_c

Out[16]:
```

	time	M_0	M_1
0	0.0	100.0	200.0
1	1.0	400.0	300.0
2	2.0	850.0	950.0
3	3.0	2275.0	2175.0

```

In [17]: M_c_1=np.array([[1],[1]])
df_L_c_1=leslie_calc(L_c,M_c_1,6)
df_L_c_1

Out[17]:
```

	time	M_0	M_1
0	0.0	1.000000	1.000000
1	1.0	2.500000	2.500000
2	2.0	6.250000	6.250000
3	3.0	15.625000	15.625000
4	4.0	39.062500	39.062500
5	5.0	97.656250	97.656250
6	6.0	244.140625	244.140625

```

In [18]: np.matmul(L_c,M_c_1)/2.5

Out[18]: array([[1.],
[1.]])

In [ ]:

```

2. A drug in which the amount of drug is eliminated is constant. At time $t = 0$ an amount $a_0 = 20\text{mg}$ is present in the blood. One hour later, at $t = 1$, an amount $a_1 = 14\text{mg}$ is present.

- (a) Assuming that no drug is added to the blood between $t = 0$ and $t = 1$, calculate the amount of drug that is removed from the blood each hour.

$$a_0 - a_1 = 20 - 14 = 6 \text{ 6mg are removed each interval.}$$

- (b) Write a recursion relation for the amount of drug a_t that is present at time t . Assume no extra drug is added to the blood.

$$a_{t+1} = \max(0, a_t - 6)$$

- (c) Find an explicit formula for a_t as a function of t .

$$a_t = \max(a_0 - 6t, 0) = \max(20 - 6t, 0)$$

- (d) When does the amount of drug present in the blood first drop to 0?

$$2 - 6 \cdot 3 = 2, 20 - 6 \cdot 4 = -4, \text{ so drug first reaches 0 when } t = 4.$$

3. We want to model painkillers that are absorbed into the blood from a slow release pill. Our mathematical model for the amount, a_t , of the drug in the blood t hours after the pill is taken must include the amount absorbed from the pill each hour. Assume that the amount absorbed from the pill between time t and the time $t + 1$ is $10 \cdot (0.4)^t$.

- (a) The amount of drug is eliminated depends on the amount in the system. 10% of the drug is eliminated from the blood each hour. Write down the recursion relation for a_{t+1} in terms of a_t .

$$a_{t+1} = 10 \cdot (0.4)^t + 0.90 \cdot a_t$$

- (b) Assuming that $a_0 = 0$, meaning that no drug is present in the blood initially, calculate the amount of drug present at times $t = 1, 2, \dots, 6$.

```
df=slow_release_df(0,6)
df
```

	Time (Hours)	N_t
0	0.0	0.00000
1	1.0	4.00000
2	2.0	5.20000
3	3.0	5.32000
4	4.0	5.04400
5	5.0	4.64200
6	6.0	4.21876

- (c) What is the maximum amount of drug present at any time in this interval? At what time is this maximum amount reached?

$$N_3 = 5.32 \text{ at } t = 3$$

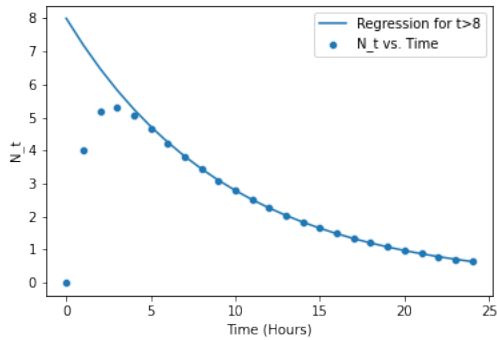
- (d) Use a computer to calculate the amount of drug present in hourly intervals from $t = 0$ up to $t = 24$.

```
[3]: df=slow_release_df(0,24)
df
```

```
[3]:
```

	Time (Hours)	N_t
0	0.0	0.000000
1	1.0	4.000000
2	2.0	5.200000
3	3.0	5.320000
4	4.0	5.044000
5	5.0	4.642000
6	6.0	4.218760
7	7.0	3.813268
8	8.0	3.436495
9	9.0	3.097267
10	10.0	2.788589
11	11.0	2.510149
12	12.0	2.289302
13	13.0	2.033439
14	14.0	1.830122
15	15.0	1.647120
16	16.0	1.482413
17	17.0	1.334173
18	18.0	1.200757
19	19.0	1.080681
20	20.0	0.972613
21	21.0	0.875352
22	22.0	0.787817
23	23.0	0.709035
24	24.0	0.638132

(e) Show that when t is large, the amount of drug present in the blood decreases approximately exponentially with t .



```

In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: def slow_release_df(N_0,time):
df=pd.DataFrame(
    {"Time (Hours)":i[0],
     "N_t":(N_0)}
    for i in np.arange(1,time+1),
    N_0=0.1*N_0
    N_0=10*np.power(0.4,i)
    x=("Time (Hours)"+"N_t"+"N_0")
    df=df.append(x,ignore_index=True)
    return df

In [3]: df=slow_release_df(0,24)
df

Out[3]:
```

	Time (Hours)	N_t
0	0.0	0.000000
1	1.0	4.000000
2	2.0	5.200000
3	3.0	5.320000
4	4.0	5.044000
5	5.0	4.642000
6	6.0	4.218760
7	7.0	3.813268
8	8.0	3.438495
9	9.0	3.097267
10	10.0	2.788589
11	11.0	2.510149
12	12.0	2.269302
13	13.0	2.033439
14	14.0	1.830122
15	15.0	1.647120
16	16.0	1.482413
17	17.0	1.334173
18	18.0	1.200757
19	19.0	1.080681
20	20.0	0.972613
21	21.0	0.876352
22	22.0	0.787817
23	23.0	0.709035
24	24.0	0.638132

```

In [4]: df.sort_values("N_t",ascending=False).head(1)

Out[4]:
```

	Time (Hours)	N_t
3	3.0	5.32

```

In [8]: df_10=df[df["Time (Hours)"]>8]
df_exp=np.exp(np.polyfit(x=df_10["Time (Hours)"],y=np.log(df_10["N_t"]),deg=1))
sns.scatterplot(data=df,x="Time (Hours)",y="N_t",label="N_t vs. Time")
sns.lineplot(data=df,x="Time (Hours)",y=df_exp[1]*np.power(df_exp[0],df["Time (Hours)"]),label="Regression for t>8")
plt.ylabel("N_t")
plt.xlabel("Time (Hours)")
plt.savefig("Math_142_HW_2_3a")



In [ ]:

```

4. This question is about developing a model for the life-cycle of a type of nanoflagellates, a type of swimming cell that lives in water. The nanoflagellate cells go through three different life history stages. All cells start as swimming larvae (we call this the juvenile phase). They swim around looking for a surface (e.g. a plant) to settle on. When they settle they become tethered. Tethered cells feed for a while, and then when they have exhausted the food nearby they break their tethers, and start to swim around in search of another place to settle (we call these cells freely swimming). Only tethered cells are capable of reproducing. At all stages the organism can also die, in which case it won't transition to another stage. You are modeling for how the three populations vary with time: at the k th census there are $N_F^{(k)}$ freely swimming cells, $N_J^{(k)}$ juvenile cells and $N_T^{(k)}$ tethered cells. Measurements are taken daily. Our model needs to incorporate the following information:

- In each day, a fraction m of cells in all of the stages die.
- Of the freely swimming cells that do not die, a fraction t will become tethered.
- Of the tethered cells that do not die, a fraction u will become untethered (i.e. enter the freely swimming phase).
- Of the tethered cells that do not die, a fraction b will divide in two, producing a larval cell as a daughter.

- Of the juvenile cells that do not die, a fraction c will become tethered.
- (a) Show that the changes in this population from one census to another can be modeled by the following Leslie matrix:

$$\begin{pmatrix} N_F^{(k+1)} \\ N_J^{(k+1)} \\ N_T^{(k+1)} \end{pmatrix} = \begin{pmatrix} (1-m)(1-t) & 0 & (1-m)u \\ 0 & (1-m)(1-c) & (1-m)b \\ (1-m)t & (1-m)c & (1-m)(1-u) \end{pmatrix} \begin{pmatrix} N_F^{(k)} \\ N_J^{(k)} \\ N_T^{(k)} \end{pmatrix}$$

where you will need to explain the terms, and give expressions for all the terms that have been replaced by $*$.

We know that each day, a fraction m of cells in all of the stages die, so each element of the matrix will have a $(1-m)$ to model the proportion who survive. $(1-t)(1-m)$ gives the proportion of surviving freely swimming cells that stay untethered. $t(1-m)$ gives the proportion of surviving freely swimming cells that become tethered. $(1-c)(1-m)$ gives the proportion of surviving juveniles that stay juveniles. $c(1-m)$ gives the proportion of surviving juveniles that become tethered. $u(1-m)$ gives the proportion of surviving tethers that become untethered. $b(1-m)$ gives the proportion of surviving tethers that birth new juveniles. $(1-u)(1-m)$ gives the proportion of surviving tethers that stay tethered.

- (b) It is possible that, between censuses a tethered cell may divide, and then the original parent cell dies. Is this kind of transition included in our model? Can you think of any other transitions that are not included in the model?

A tethered nanoflagellate can birth a juvenile at the start of the new time interval and untether before the next census occurs. Our model doesn't include death, but this transition is close. Our model does not include the possibility of a cell maturity to a tethered nanoflagellate and also becoming untethered between intervals.

Situations like (b) are a manifestation of the general problem of what arises when the time interval between censuses is too large - individuals can perform more than one transition in a day (i.e. reproduce and then die). This is part of the reason why we will be introducing continuous time models (equivalent to models in which the interval between censuses $\Delta t \rightarrow 0$).

5. Submit the code you used for any and all of the problems. (Print pdf the code) If we have any questions, we may ask you for the actual code file.
(attached code following each problem where I used code)